

Numerické řešení obyčejných diferenciálních rovnic

Numerical solution to ordinary differential equations

Adam Bílek

Bakalářská práce

Vedoucí práce: doc. Ing. Dalibor Lukáš, Ph.D.

Ostrava, 2021

Abstrakt

Tato práce se zabývá porovnáním Runge-Kutta metod s Galerkinovou a kolokační metodou pro numerické řešení obyčejných diferenciálních rovnic a jejich soustav. K jednotlivým metodám poskytneme základní teoretické myšlenky a principy, na kterých jsou vystavěny. Dále uvedeme jejich algoritmizaci pro praktickou implementaci. A nakonec efektivitu jednotlivých metod otestujeme na reálných lineárních i nelineárních příkladech.

Klíčová slova

Obyčejné diferenciální rovnice, Numerické řešení ODR, Nelineární ODR, Galerkinova metoda, Kolokační metoda, Implicitní metody Runge-Kutta, Explicitní metody Runge-Kutta

Abstract

This work deals with the comparison of Runge-Kutta methods with Galerkin and collocation method for numerical solution of ordinary differential equations and their systems. We will provide basic theoretical ideas and principles on which these methods rely. Next, we present their algorithmization for practical implementation. Finally, we test the effectiveness of individual methods on real linear and nonlinear examples.

Keywords

Ordinary differential equations, Numerical solution of ODEs, Nonlinear ODEs, Galerkin method, Collocation method, Implicit Runge-Kutta methods, Explicit Runge-Kutta methods

Obsah

Seznam použitých symbolů a zkratek	4
1 Úvod	5
2 Motivace	6
3 Teorie k řešení ODR	8
4 Metody Runge-Kutta	10
4.1 Explicitní Runge-Kutta	10
4.2 Implicitní Runge-Kutta	12
4.3 Runge-Kutta pro soustavy ODR	14
5 Galerkinova metoda	16
5.1 Formulace slabého řešení	16
5.2 Galerkinova aproximace	18
6 Algoritmizace	21
6.1 Explicitní Runge-Kutta	21
6.2 Implicitní Runge-Kutta	22
6.3 Galerkinova metoda	23
6.4 Kolokační metoda	27
7 Numerické experimenty	30
7.1 Ukázka jednotlivých metod	30
7.2 Ověření řádů konvergence	33
7.3 Řešení soustav ODR	36
8 Závěr	42
Literatura	43

Seznam použitých zkratek a symbolů

\mathbb{R}	– Množina všech reálných čísel
\mathbb{R}^n	– n -rozměrný euklidovský prostor
\mathbb{N}	– Množina všech přirozených čísel
$y'(x)$	– První derivace funkce y podle proměnné x
$y^{(n)}(x)$	– n -tá derivace funkce y podle proměnné x
$\dim(X)$	– Dimenze prostoru X
$\ \cdot\ _X$	– Norma na prostoru X
$\langle \cdot, \cdot \rangle$	– Skalární součin na prostoru X
X^*	– Duální prostor k prostoru X
$\langle A, b \rangle$	– Hodnota funkcionálu A v bodě b
$\text{Ker}(A)$	– Jádro A
$\text{Ker}(A)^\perp$	– Množina prvků ortogonálních k jádru A
ODR	– Obyčejná diferenciální rovnice
ERK	– Explicitní Runge-Kutta
IRK	– Implicitní Runge-Kutta
DIRK	– Diagonal implicit Runge-Kutta
SDIRK	– Singly-diagonal implicit Runge-Kutta

1. Úvod

Tato práce se zabývá popsáním, algoritmizací a testováním několika numerických metod pro řešení obyčejných diferenciálních rovnic. Konkrétně se jedná o metody explicitní a implicitní Runge-Kutta, Galerkinova metoda a metoda kolokační.

Obyčejné diferenciální rovnice slouží typicky k popisování některých časově závislých jevů téměř ve všech vědních oborech. Jedná se o rovnice, ve kterých hledaná neznámá je funkce závislá právě na jedné nezávislé proměnné. I přesto, že mnohé z nich umíme řešit analyticky, ve srovnání s počtem, které řešit analyticky neumíme, je toto množství zanedbatelné. Proto se v praxi uchylujeme k použití nejrůznějších numerických metod, které nám umožňují řešit tyto rovnice alespoň přibližně.

V současnosti jsou jedny z nejpoužívanějších metod tzv. metody Runge-Kutta. Jedná se o jednokrokové metody, tzn. neudrží žádnou informaci o chování funkce na již vyřešeném úseku. Jednokrokové metody spočívají v lokální aproximaci Taylorova polynomu k výpočtu následujícího kroku, takto postupně aproximují celé řešení na požadovaném intervalu.

Kromě Runge-Kutta metod se zaměříme také na méně známé metody pro řešení obyčejných diferenciálních rovnic, konkrétně metodu Galerkinovu a kolokační. Tyto metody spočívají v aproximaci řešení pomocí nějakých známých bázeových funkcí na celém intervalu řešení. Díky tomu tyto metody vykazují extrémně rychlou konvergenci, která by šla využít např. při výpočtech s vysokou přesností.

Jako první se v kapitole 2 podíváme detailněji na motivaci k řešení obyčejných diferenciálních rovnic, jak takové numerické řešení může vypadat, a motivaci ke vzniku této práce. Dále si v kapitole 3 představíme definici diferenciálních rovnic a jejich soustav, a uvedeme některé základní existenční věty. V kapitole 4 představíme metody Runge-Kutta a jejich odvození a vlastnosti. Dále si v této kapitole zavedeme pojem stabilita, který by měl ovlivňovat volbu metody pro řešení konkrétního problému. V kapitole 5 odvodíme Galerkinovu metodu a uvedeme základní věty prokazující jak funkčnost, tak rychlost konvergence metody. Poté v kapitole 6 ukážeme jak tyto metody algoritmizovat, aby byly použitelné při praktických výpočtech. Nakonec v kapitole 7 všechny metody uvedené v této práci otestujeme na praktických úlohách a empiricky pro tyto metody ověříme řády konvergence.

2. Motivace

Uvažujme netlumené kmitání matematického kyvadla. Tento jev můžeme popsat obyčejnou diferenciální rovnicí

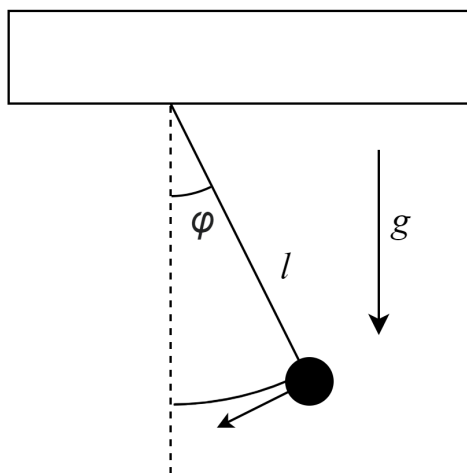
$$\varphi''(t) + \frac{g}{l} \sin \varphi(t) = 0$$

kde $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ reprezentuje neznámou funkci vychýlení kyvadla v závislosti na čase a parametry $g > 0$ a $l > 0$ reprezentují fyzikální vlastnosti konkrétního jevu. Pokud k naší rovnici přidáme ještě počáteční podmínky

$$\varphi(t_0) = \varphi_0$$

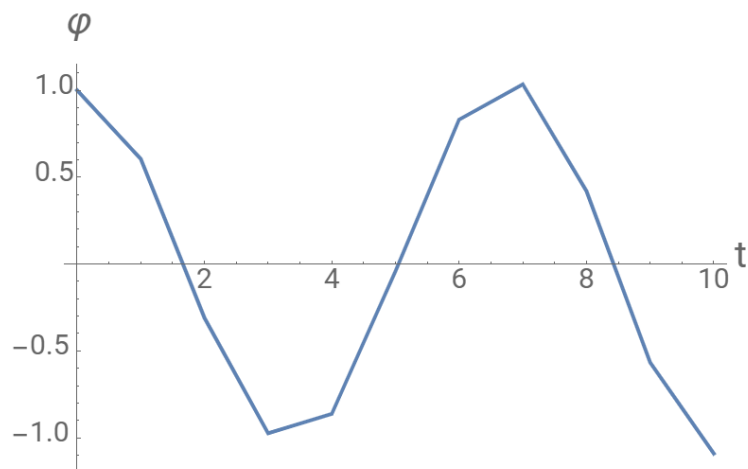
$$\varphi'(t_0) = \varepsilon_0$$

vznikne Cauchyova úloha, kde φ_0 reprezentuje počáteční vychýlení ramene kyvadla v nějakém počátečním čase t_0 a ε_0 reprezentuje počáteční úhlovou rychlost ramene kyvadla v stejném počátečním čase t_0 .



Obrázek 2.1: Matematické kyvadlo

Numerické řešení takovéto úlohy bude většinou znamenat nalezení $\varphi(T) = \varphi_T$ pro nějaké námi zadané $T > t_0$. Tento výsledek by se dal interpretovat jako vychýlení ramene kyvadla v čase T . To však nemusí být jediný bod našeho zájmu. Můžeme se například zajímat o průběh funkce $\varphi(t)$ na celém intervalu $\langle t_0, T \rangle$, nebo kde funkce $\varphi(t)$ na tomto intervalu nabývá lokálních extrémů apod.



Obrázek 2.2: Příklad numerického řešení kyvadla pro $t_0 = 0$, $\varphi_0 = 1$ a $\varepsilon_0 = 0$ na intervalu $\langle 0, 10 \rangle$

Hlavní motivace ke vzniku této práce tedy je poskytnout přehled několika známých i méně známých metod pro numerické řešení takovýchto úloh. Konkrétně poskytnout základní teoretické myšlenky na pozadí těchto metod a s nimiž související vlastnosti. Dále jednotlivé metody principiálně algoritmizovat, aby bylo možné jejich empirické testování. A nakonec ukázat a porovnat jednotlivé metody na praktických příkladech.

3. Teorie k řešení ODR

V první kapitole se zaměříme na definici základních pojmů a ukážeme si věty, které budeme využívat ve zbytku práce.

Začneme obyčejnou diferenciální rovnicí prvního řádu.

Definice 3.1 *Obyčejnou diferenciální rovnici prvního řádu budeme rozumět*

$$y'(x) = f(x, y(x)), \quad (3.1)$$

kde

$f : \mathbb{R}^2 \rightarrow \mathbb{R}$ je zadaná spojitá funkce

$y : \mathbb{R} \rightarrow \mathbb{R}$ je neznámá funkce proměnné $x \in \mathbb{R}$

Přidáme-li k (3.1) počáteční podmínku $y(x_0) = y_0$ dostáváme tzv. Cauchyovou úlohu.

Věta 3.1 *Nechť $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ je spojitá funkce definovaná na otevřené množině $\Omega \subset \mathbb{R}^2$, $(x_0, y_0) \in \Omega$. Nechť $\frac{\partial f}{\partial y}$ je navíc spojitá na Ω . Potom Cauchyova úloha má právě jedno maximální řešení.*

Tato věta je jedna z mnoha existenčních vět. Další a mnohdy obecnější věty i s důkazy nalezneme např. v [1]. Samotná však nepřináší žádnou informaci o velikosti intervalu, na kterém je garantovaná existence řešení. Z tohoto důvodu si uvedeme následující větu.

Věta 3.2 *Nechť funkce $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ je spojitá na obdélníku $O = \langle x_0 - a, x_0 + a \rangle \times \langle y_0 - b, y_0 + b \rangle$, kde $a > 0$, $b > 0$. Nechť existuje číslo $L \in \mathbb{R}$ takové, že pro každé dva body $(x, y_1) \in O$, $(x, y_2) \in O$ platí $|f(x, y_1) - f(x, y_2)| \leq L|y_1 - y_2|$. Dále označme M jako maximální hodnotu funkce f na obdélníku O a předpokládejme $M > 0$.*

Potom existuje právě jedna funkce $\varphi(x)$ definovaná na intervalu $(x_0 - \alpha, x_0 + \alpha)$, kde $\alpha = \min\{a, \frac{b}{M}\}$, která je na tomto intervalu řešením Cauchyovy úlohy.

V praxi se však často setkáme se složitějšími diferenciálními rovnicemi, proto si zadefinujeme také soustavu n obyčejných diferenciálních rovnic m -tého řádu.

Definice 3.2 *Soustavou n obyčejných diferenciálních rovnic m -tého řádu budeme rozumět*

$$y^{(m)}(x) = f(x, y(x), y'(x), \dots, y^{(m-1)}(x)), \quad (3.2)$$

kde

$f : \mathbb{R}^{1+n \times m} \rightarrow \mathbb{R}^n$ je zadaná funkce

$y : \mathbb{R} \rightarrow \mathbb{R}^n$ je neznámá funkce proměnné $x \in \mathbb{R}$

Obdobné věty jako pro (3.1) lze formulovat i zde, avšak zabývat se takto obecnou formou může být v mnoha ohledech nepraktické.

Pozorování 3.1 *Mějme soustavu diferenciálních rovnic (3.2) a provedme substituci*

$$y'(x) = y_1(x), \quad y'_1(x) = y_2(x), \quad \dots, \quad y'_{m-2}(x) = y_{m-1}(x)$$

dostáváme

$$y'_{m-1}(x) = f(x, y(x), y_1(x), \dots, y_{m-1}(x))$$

$$y'_{m-2}(x) = y_{m-1}(x)$$

$$\vdots$$

$$y'_1(x) = y_2(x)$$

$$y'(x) = y_1(x)$$

všimněme si, že výsledný výraz je soustava $n \times m$ diferenciálních rovnic prvního řádu.

Díky tomuto pozorování se můžeme zaměřit pouze na řešení diferenciálních rovnic prvního řádu, aniž bychom se připravili o schopnost řešit rovnice řádů vyšších.

4. Metody Runge-Kutta

Asi jedny z nejznámějších metod pro řešení obyčejných diferenciálních rovnic jsou metody z rodiny Runge-Kutta. Jedná se o jednokrokové metody tzn., že si neudržují informaci o průběhu řešení z předchozích kroků. Tyto metody spočívají v lokální aproximaci Taylorova polynomu k výpočtu přibližného řešení. Více k historii a vývoji Runge-Kutta metod nalezneme v [2]. Tyto metody se dělí na dvě základní skupiny, konkrétně explicitní a implicitní Runge-Kutta metody. Rozdíly v těchto skupinách si popíšeme v následujících sekcích.

4.1 Explicitní Runge-Kutta

Nyní odvodíme nejjednodušší metodu z rodiny explicitních Runge-Kutta metod.

Vezměme počáteční úlohu

$$y'(x) = f(x, y(x)) \quad y_0 = y(x_0) \quad (4.1)$$

Řekněme, že nás bude zajímat hodnota $y(x_0 + h)$. Pokud je h dostatečně malé, pak můžeme využít nástrojů matematické analýzy, v tomto případě Taylorova polynomu

$$y(x_0 + h) \approx y(x_0) + h \cdot y'(x_0) = y(x_0) + h \cdot f(x_0, y(x_0)) = y_0 + h \cdot f(x_0, y_0)$$

Pokud tento postup zobecníme tak, že budeme chtít znát hodnotu y v bodě $x_T > x_0$ a jsme ochotni udělat n kroků, pak

$$h = \frac{x_T - x_0}{n} \quad y_{i+1} = y_i + h \cdot f(x_0 + h \cdot i, y_i) \quad i \in \{1, \dots, n-1\}$$

potom dostáváme hodnotu $y_n \approx y(x_T)$. Tomuto postupu se říká Eulerova metoda s neměnným krokem a zároveň se jedná o explicitní Runge-Kutta metodu prvního řádu.

Definice 4.1 Mějme rovnici (4.1), pak definujeme n stavovou explicitní metodu Runge-Kutta jako

$$y_{k+1} = y_k + h \sum_{i=1}^n b_i z_i$$

$$z_i = f(x_k + c_i h, y_k + h \sum_{j=1}^{i-1} a_{i,j} z_j), \quad i = 1, \dots, n$$

Koeficienty a, b, c určují metodu a typicky se zapisují do Butcherovy tabulky

c_1					
c_2	$a_{2,1}$				
c_3	$a_{3,1}$	$a_{3,2}$			
\vdots	\vdots		\ddots		
c_n	$a_{n,1}$	$a_{n,2}$	\cdots	$a_{n,n-1}$	
	b_1	b_2	\cdots	b_{n-1}	b_n

Dále řekneme, že metoda má řád konvergence p pokud je chyba v jednom kroku řádu $\mathcal{O}(h^{p+1})$.

Obecně platí, že pro definici (4.1) $n \geq p$, konkrétně pro dosažení řádu konvergence $p = 1, \dots, 6$ je minimální potřebný počet stavů metody $n = 1, 2, 3, 4, 6, 7$ respektive, viz. [2]. Metody s vyšším řádem konvergence také existují, nebylo však dokázáno, že by se stejného řádu konvergence nedalo dosáhnout pomocí méně stavové metody.

Vraťme se nyní zpět k Eulerově metodě a uvažujme příklad

$$y'(x) = \lambda y(x) \quad y_0 = y(0) = 1 \quad (4.2)$$

Analytickým řešením této rovnice je $y(x) = e^{\lambda x}$, pokud však budeme tuto rovnici řešit pomocí explicitní Eulerovy metody dostaneme předpis

$$y_{n+1} = y_n + h\lambda y_n \Rightarrow y_n = (1 + h\lambda)^n$$

všimněme si, že pokud bude $\lambda < 0$ a $|1 + h\lambda| > 1$, pak se numerické řešení s každým krokem exponenciálně vzdaluje od řešení analytického. Této vlastnosti říkáme nestabilita.

Definice 4.2 Mějme rovnici (4.2) a použijme Runge-Kutta metodu, pak lze vyjádřit $y_n = \phi(\lambda h)^n y_0$, kde funkci $\phi(\lambda h)$ se říká funkce stability, která zároveň slouží k definici regionu stability jako množiny

$$S_r = \{\forall z \in \mathbb{C} : |\phi(z)| < 1\}$$

Dále, pokud platí

$$S_r \subseteq \{\forall z \in \mathbb{C} : \operatorname{Re}(z) < 0\}$$

řekneme, že metoda je A-stabilní. Pokud navíc platí

$$\lim_{z \rightarrow \pm\infty} \phi(z) = 0$$

řekneme, že metoda je L-stabilní.

A-stabilní metoda by tedy měla na modelové rovnici vykazovat následující chování

$$\forall h > 0, \forall \lambda < 0 : \lim y_n = 0$$

Tuto vlastnost však nemá žádná explicitní Runge-Kutta metoda [2].

4.2 Implicitní Runge-Kutta

Motivací ke zkoumání těchto metod je fakt uvedený na konci předchozí sekce. Protože různé diferenciální rovnice vykazují různé chování, setkáme se často s diferenciálními rovnicemi, které mají „velký záporný λ faktor“. Těmto rovnicím se říká tuhé diferenciální rovnice (z anglického *stiff equation*) a k jejich řešení je vhodné použít právě A-stabilní, nebo L-stabilní metody.

Vezměme opět počáteční úlohu (4.1) opět provedeme odvození Eulerovy metody, tentokrát však použijeme Taylorův polynom sestavený v bodě $y(x_0 + h)$ namísto $y(x_0)$

$$y(x_0) \approx y(x_0 + h) - h \cdot y'(x_0 + h) \Rightarrow y(x + h) = y_0 + h \cdot f(x_0 + h, y(x + h))$$

Zobecněním dostáváme metodu

$$h = \frac{x_T - x_0}{n}, \quad y_{i+1} = y_i + h \cdot f(x_0 + h \cdot (i+1), y_{i+1}), \quad i \in \{1, \dots, n-1\}$$

Této metodě se říká implicitní Eulerova metoda, protože namísto explicitně vyjádřeného y_{i+1} jako tomu bylo v případě explicitních metod, je zde potřeba k výpočtu y_{i+1} obecně řešit nelineární rovnici.

Definice 4.3 *Mějme rovnici (4.1), pak definujeme n stavovou implicitní Runge-Kutta jako*

$$y_{k+1} = y_k + h \sum_{i=1}^n b_i z_i$$

$$z_i = f(x_k + c_i h, y_k + h \sum_{j=1}^n a_{i,j} z_j), \quad i = 1, \dots, n$$

Koeficienty a, b, c určují metodu a typicky se zapisují do Butcherovy tabulky

c_1	$a_{1,1}$	$a_{1,2}$	\cdots	$a_{1,n}$
c_2	$a_{2,1}$	$a_{2,2}$	\cdots	$a_{2,n}$
c_3	$a_{3,1}$	$a_{3,2}$	\cdots	$a_{3,n}$
\vdots	\vdots	\vdots	\ddots	\vdots
c_n	$a_{n,1}$	$a_{n,2}$	\cdots	$a_{n,n}$
	b_1	b_2	\cdots	b_n

Dále řekneme, že metoda má řád konvergence p pokud je chyba v jednom kroku řádu $\mathcal{O}(h^{p+1})$.

Z definice implicitní Runge-Kutta metody vyplývá, že obecně je potřeba v každém kroku řešit soustavu nelineárních rovnic. To dělá tyto metody řádově složitější na praktické použití. Na druhou stranu je tato nevýhoda kompenzovaná jinými vlastnostmi, například Gauss-Legendreovy [3] metody mají řád konvergence $p = 2n$, takže můžeme konstruovat metody s libovolným řádem konvergence, u níž zároveň víme počet stavů.

Dále se zaměříme na stabilitu implicitních Runge-Kutta metod, vezmeme modelovou rovnici (4.2) s analytickým řešením $y(x) = e^{\lambda x}$. Nyní však budeme tuto rovnici řešit pomocí implicitní Eulerovy metody

$$y_{n+1} = y_n + h\lambda y_{n+1} \Rightarrow y_n = \left(\frac{1}{1 - h\lambda} \right)^n$$

Region stability pro tuto metodu splňuje

$$S_{r-euler} = \{ \forall z \in \mathbb{C} : \left| \frac{1}{1 - z} \right| < 1 \} \subseteq \{ \forall z \in \mathbb{C} : \operatorname{Re}(z) < 0 \}$$

tudíž je metoda A-stabilní. Navíc metoda splňuje i

$$\lim_{z \rightarrow \pm\infty} \left| \frac{1}{1 - z} \right| = 0$$

a proto je metoda také L-stabilní. Toto tvrzení zároveň dokazuje, že existují implicitní A-stabilní a L-stabilní Runge-Kutta metody, takže některé z nich jsou vhodné i pro řešení tuhých diferenciálních rovnic.

4.3 Runge-Kutta pro soustavy ODR

Na obyčejných diferenciálních rovnicích se snadno demonstrovaly základní principy, ale v praxi se mnohem častěji setkáváme se soustavami diferenciálních rovnic. Jak už jsme uvedli v kapitole 3, můžeme do soustav také převést rovnice vyšších řádů, a proto se v této sekci budeme věnovat zobecnění metod pro řešení jedné diferenciální rovnice na řešení jejich soustav.

V případě explicitních Runge-Kutta metod je to celkem přímočarý postup. Vezměme definici (4.1) a nahradíme skalární prvky za vektorové

$$y = (y_1, \dots, y_m) \quad f(x, y) = (f_1(x, y_1, \dots, y_m), \dots, f_m(x, y_1, \dots, y_m))$$

Automaticky dostáváme explicitní metody Runge-Kutta pro řešení soustav

$$y_{s,k+1} = y_{s,k} + h \sum_{i=1}^n b_i z_{s,i}$$

$$z_{s,i} = f_s(x_k + c_i h, y_k + h \sum_{j=1}^{i-1} a_{i,j} z_j), \quad s = 1, \dots, m$$

Komplexita metody se sice zvýšila, ale stále se jedná jen o dosazení do vzorce.

Na druhou stranu, pokud toto zobecnění použijeme pro metody implicitní (4.3), dostáváme se do oblasti řešení obecně velmi komplikovaných soustav nelineárních rovnic. V praxi se snažíme složitost těchto soustav zmírnit použitím vhodných metod, například metody DIRK (diagonal implicit Runge-Kutta) [4] a SDIRK (singly-diagonal implicit Runge-Kutta) nám umožní řešit postupně několik menších soustav v jednom kroku namísto jedné velké, i přesto je však efektivní implementace implicitních Runge-Kutta metod pro řešení rozsáhlých nelineárních soustav diferenciálních rovnic velmi náročný problém. Více k teorii Runge-Kutta metod a obecně numerickému řešení ODR najdeme v [5].

5. Galerkinova metoda

V předchozí kapitole jsme si rozebrali jednokrokové metody pro numerické řešení obyčejných diferenciálních rovnic. Jedna z nevýhod těchto metod byla, že chyba aproximace rostla s počtem kroků, a druhá, že pro dosažení vyššího řádu konvergence byla zapotřebí „lepší metoda“. Nyní si však ukážeme metodu, která nejen zvyšuje svůj řád konvergence s počtem uzlů, ale navíc tímto tempem konverguje globálně na celém zadaném intervalu. Zároveň se tak jedná o metodu vhodnou pro výpočty s vysokou přesností.

Přistupme tedy k problému hledání numerického řešení diferenciální rovnice jako k problému aproximace funkce, která je přibližným řešením oné rovnice, pomocí funkce složené ze známých bázových funkcí. Uvažujme tedy lineární diferenciální rovnici

$$y'(x) + f(x)y(x) = g(x) \quad y(0) = y_0 \quad (5.1)$$

kde funkce f a g jsou zadané hladké funkce. Řekněme, že si vystačíme s aproximací hledané funkce y polynomem

$$y_n(x) = y_0 + \sum_{i=1}^n \alpha_i x^i$$

takovou, aby koeficienty α minimalizovaly nějakou normu

$$\arg \min_{\alpha} \|y - y_n\|$$

Myšlenka je sice jasná, ale bez dalších poznatků je provedení obtížné až nemožné.

5.1 Formulace slabého řešení

Pokud se ještě jednou vrátíme k rovnici (5.1), můžeme si všimnout, že rovnice dává smysl pouze, pokud je y diferencovatelné na $(0, T)$. Tomuto se říká silná formulace diferenciální rovnice. Naneštěstí je tato formulace příliš silná v tom smyslu, že $f(x)$ a $g(x)$ musejí být alespoň spojité. Tento fakt nám znemožňuje řešit některé fyzikální problémy, a proto zavádíme slabou variační formulaci, která má daleko nižší požadavky na funkce $f(x)$ a $g(x)$ i $y(x)$, a která nám umožňuje hledat řešení daleko „komplexnější“, než jaké nám umožňovala formulace silná.

Uvažujme tedy diferenciální rovnici pro funkci y definovanou na intervalu $\langle 0, T \rangle$

$$y'(x) + y(x) = f(x), \quad y(0) = y_0$$

V klasické formulaci bychom hledali řešení

$$y \in C^0(\langle 0, T \rangle) \cap C^1((0, T))$$

Pro slabou formulaci zapíšeme $y(x) = y_h(x) + y(0)$ a hledáme

$$\begin{aligned} y_h(x) &\in V = H_{0,}^1((0, T)) \\ \forall q \in L^2((0, T)) : \int_0^T (y_h'(x) + y_h(x))q(x)dx &= \int_0^T (f(x) - y_0)q(x)dx \end{aligned} \quad (5.2)$$

kde

$$\begin{aligned} L^2((0, T)) &= \{v : (0, T) \rightarrow \mathbb{R} : v \text{ je lebesgueovsky měřitelná a } \int_0^T v^2(x)dx < \infty\} \\ H_{0,}^1((0, T)) &= \{w : w \in L^2((0, T)) \wedge w' \in L^2((0, T)) \wedge w(0) = 0\} \end{aligned}$$

Integrály jsou Lebesgueovy a derivace jsou ve smyslu distribucí.

Věta 5.1 *Mějme Hilbertovy prostory V, Q , prostor spojitých lineárních funkcíonálů nad Q ozn. Q^* (v našem případě $H_0^1((0, T)) = V \subset Q = Q^* = L^2((0, T))$), dualitní párování $\langle \varphi, q \rangle_{Q^* \times Q}$ (v našem případě $\langle \varphi, q \rangle_{Q^* \times Q} = \int_0^T \varphi(x)q(x)dx$), spojitý lineární operátor $A \in \mathcal{L}(V, Q^*)$ a pravou stranu A ozn. b . Platí-li tzv. inf-sup podmínka*

$$\exists c > 0, \forall v \in \text{Ker}(A)^\perp : \|v\|_V \leq \sup_{0 \neq q \in Q} \frac{\langle Av, q \rangle_{Q^* \times Q}}{\|q\|_Q}$$

pak

$$\exists! y_h \in V : y_h \text{ je řešením (5.2) a } \|y_h\|_V \leq \frac{1}{c} \|b\|_{Q^*} \text{ je stabilní.}$$

Důkaz této věty nalezneme v [6, Theorem 3.7].

Pomocí pojmů zdefinovaných v předchozí větě lze (5.2) zapsat jako abstraktní úlohu, ve které hledáme $u \in V$ takové, že

$$\forall q \in Q : \langle Au, q \rangle_{Q^* \times Q} = \langle b, q \rangle_{Q^* \times Q}$$

Za splnění jistých předpokladů má tato úloha právě jedno stabilní řešení. Pokud je toto řešení navíc dostatečně hladké, je rovno řešení klasickému.

5.2 Galerkinova aproximace

V předchozí sekci jsme zformulovali slabé řešení, to však ještě nestačí k numerickému vyřešení diferenciální rovnice pomocí Galerkinovy metody. Princip této metody spočívá ve volbě konečně-rozměrných podprostorů $V_n \subset V$ a $Q_m \subset Q$, kde n a m označují dimenze, k vyřešení slabé formulace diferenciální rovnice. Nově tedy řešíme úlohu nalezení $u_n \in V_n$ takového, že

$$\forall q_m \in Q_m : \langle Au_n, q_m \rangle_{Q^* \times Q} = \langle b, q_m \rangle_{Q^* \times Q}$$

Podle volby báze funkcí dělíme Galerkinovu metodu na dva typy:

1. Metoda konečných prvků

Ta spočívá ve volbě vhodných uzlů x_i na intervalu $\langle 0, T \rangle$, báze funkce jsou pak voleny tak, aby byly navzájem spojité a na intervalech mezi všemi sousedními uzly lineární, nebo nulové, např.:

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x \in [x_{i-1}, x_i] \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x \in [x_i, x_{i+1}] \\ 0, & x \notin [x_{i-1}, x_{i+1}] \end{cases}$$

2. Spektrální metoda

Spočívá ve volbě analytických funkcí tvořících báze vektorových prostorů, jenž jsou nenulové téměř všude, např. monomiály, Legendreovy polynomy, Chebyshevovy polynomy apod.

Obě metody konvergují k řešení s rostoucím n ($\exists C > 1, \forall v_n \in V_n : \|u - u_n\|_V \leq C \|u - v_n\|_V$). V případě metody konečných prvků se jedná o lineární konvergenci

$$\min_{v_n \in V_n} \|u - v_n\|_V \leq c_u n^{-1}$$

V případě spektrální metody dostáváme tzv. spektrální rychlost konvergence. Pro jednoduchost si ji uvedeme pro případ, kdy bychom aproximovali řešení pomocí Legendreových polynomů.

Zdefinujme tedy Sobolevův prostor

$$H^k((0, T)) = \{w : \langle 0, T \rangle \rightarrow \mathbb{R} : \frac{d^i w(x)}{dx^i} \in L^2((0, T)), 0 \leq i \leq k\}$$

se skalárním součinem

$$\langle f, g \rangle_{H^k((0, T))} = \sum_{i=0}^k \int_0^T \frac{d^i f(x)}{dx^i} \frac{d^i g(x)}{dx^i} dx$$

a normou

$$\|f\|_{H^k((0, T))} = \langle f, f \rangle_{H^k((0, T))}^{1/2}$$

Dále označme Legendreovy polynomy $\{P_n(x)\}$. Ortogonální projekce libovolné funkce $f(x) \in L^2((0, T))$ je

$$P_N f(x) = \sum_{i=0}^N \hat{f}_i P_i(x), \quad \hat{f}_i = \frac{\langle f, P_i \rangle_{L^2}}{\|P_i\|_{L^2}^2}$$

Věta 5.2 *Pro libovolnou funkci $f(x) \in L^2((0, T))$ a $N \in \{0, 1, 2, \dots\}$ platí*

$$\int_0^T (f - P_N f) P dx = \langle f - P_N f, P \rangle_{L^2((0, T))} = 0, \quad \forall P \in \{P_n(x)\}$$

Důkaz nalezneme v [7, Theorem 3.4]. Z předchozí věty plyne, že chyba ortogonální projekce libovolné funkce je ortogonální na prostor ortogonálních polynomů (v našem případě Legendreových polynomů). Pokud tedy budeme aproximovat řešení naší úlohy v prostoru s bází tvořenou ortogonálními polynomy, pak bude splňovat rovnost uvedenou v předchozí větě. Můžeme tedy k určení rychlosti konvergence využít následující věty.

Věta 5.3 *Pro libovolnou funkci $f(x) \in H^p((0, T))$, $p \geq 0$, existuje konstanta C nezávislá na N taková, že platí*

$$\|f - P_N f\|_{L^2((0, T))} \leq C N^{-p} \|f\|_{H^p((0, T))}$$

Důkaz nalezneme v [7, Theorem 3.6]. Dále si všimněme, že pokud je $f(x)$ analytická, pak je řád konvergence její ortogonální projekce vyšší než libovolný řád algebraický a dostáváme

$$\|f - P_N f\|_{L^2((0, T))} \sim C e^{-\alpha N} \|f\|_{H^p((0, T))}$$

Tudíž pro analytické funkce dostáváme exponenciální konvergenci.

Nakonec uvedeme, jak bychom k řešení lineární diferenciální rovnice přistoupili v praxi. Mějme lineární ODR

$$y'(x) + f(x)y(x) = g(x), \quad y(0) = y_0$$

Řešení tedy budeme hledat jako $y(x) = y_h(x) + y_0$, kde $y_h(x)$ budeme hledat v nějakém n -rozměrném vektorovém prostoru V_n s bází $\langle \phi_1(x), \dots, \phi_n(x) \rangle$ ortogonální taktéž na V_n . Protože V_n je vektorový prostor, dá se libovolný prvek z něj zapsat jako lineární kombinace bázových funkcí. Pokud tedy chceme nalézt řešení ortogonální ke všem prvkům v prostoru V_n , stačí aby bylo ortogonální vůči všem bázovým funkcím (kterých je na rozdíl od prvků konečně mnoho, v našem případě n) takto

$$\forall i \in \{1, \dots, n\} :$$

$$\langle y'_n - f y_n, \phi_i \rangle = \langle g, \phi_i \rangle$$

$$\int_0^T (y'_n(x) + f(x)y_n(x))\phi_i(x)dx = \int_0^T g(x)\phi_i(x)dx$$

Jelikož y_n je prvkem V_n , dá se také zapsat jako lineární kombinace bázových funkcí

$$y_n(x) = \sum_{j=1}^n \alpha_j \phi_j(x)$$

Dostáváme tedy

$$\forall i \in \{1, \dots, n\} :$$

$$\int_0^T \left(\sum_{j=1}^n \alpha_j \phi'_j(x) + f(x) \sum_{j=1}^n \alpha_j \phi_j(x) \right) \phi_i(x) dx = \int_0^T g(x) \phi_i(x) dx$$

Protože jsme vycházeli z lineární diferenciální rovnice, můžeme koeficienty α vypočítat následovně

$$\forall i \in \{1, \dots, n\} :$$

$$\sum_{j=1}^n \alpha_j \left(\int_0^T (\phi'_j(x) + f(x)\phi_j(x))\phi_i(x)dx \right) = \int_0^T g(x)\phi_i(x)dx$$

$$\mathbf{A}_{i,j} = \int_0^T (\phi'_j(x) + f(x)\phi_j(x))\phi_i(x)dx$$

$$\mathbf{b}_i = \int_0^T g(x)\phi_i(x)dx$$

$$\mathbf{A}\alpha = \mathbf{b}$$

Vyřešením této soustavy lineárních rovnic obdržíme ony koeficienty α určující řešení y_n .

6. Algoritmizace

V předchozí části jsme si uvedli některé metody pro numerické řešení ODR a dále se zaměříme na jejich algoritmickou realizaci. V případě Runge-Kutta metod se budeme rovnou věnovat řešení nelineárních soustav ODR, zatímco v případě Galerkinovy metody si ukážeme algoritmické řešení jak pro lineární, tak nelineární soustavy zvlášť.

6.1 Explicitní Runge-Kutta

V případě ERK metod se jedná o celkem přímočarý postup. Můžeme využít přímo definici (4.1) a vytvořit algoritmus, který na vstup přijme funkci f na pravé straně diferenciální rovnice $y'(x) = f(x, y(x))$, $y_0 = y(x_0)$, Butcherovu tabulku v podobě vektorů $b = (b_1, b_1, \dots, b_n)$, $c = (c_1, c_1, \dots, c_n)$ a matice $A = ((a_{1,1}, \dots, a_{1,n}), (a_{2,1}, \dots, a_{2,n}), \dots, (a_{n,1}, \dots, a_{n,n}))$ a hodnot x_0, x_T , $y_0 = (y_{0,1}, y_{0,2}, \dots, y_{0,n})$ a počet kroků n . A vypočte přibližnou hodnotu $y(x_T)$.

Algoritmus 1: Explicit Runge-Kutta

Result: yn

$h \leftarrow \frac{x_T - x_0}{n}$

$yn \leftarrow y_0$

for $i = 1..n$ **do**

$xn \leftarrow x_0 + (i - 1)h$

for $j = 1..n$ **do**

$z_j \leftarrow f(xn + c_j h, yn + h \sum_{k=1}^{j-1} a_{j,k} z_k)$

end

$yn \leftarrow yn + h \sum_{j=1}^n b_j z_j$

end

Pokud bychom chtěli získat výsledek s předem specifikovanou přesností, máme k dispozici několik možností. Nejjednodušší, avšak ne moc efektivní, je opakovat celý výpočet několikrát za sebou vždy s dvojnásobným počtem kroků, dokud rozdíl po sobě jdoucích výsledků není menší než námi specifikované ϵ . Sofistikovanější metody, jako například adaptivní řízení velikosti kroku, nalezneme v [8].

6.2 Implicitní Runge-Kutta

Definice IRK se sice podobá metodě explicitní, rozdíl ve výpočtu jednotlivých kroků je však značný. Zde totiž, jak už bylo zmíněno v sekci 4.3, je zapotřebí pro každý krok vyřešit obecně soustavu nelineárních rovnic. Tento fakt dělá výpočet řádově dražší, než tomu bylo v případě metody explicitní. V praxi je tedy zapotřebí zvážit, zda se nám pro daný problém skutečně vyplatí použít tuto metodu.

Samotný algoritmus opět vychází převážně z definice, na vstup přijme funkci f na pravé straně diferenciální rovnice $y'(x) = f(x, y(x))$, $y_0 = y(x_0)$, Butcherovu tabulku v podobě vektorů $b = (b_1, b_1, \dots, b_n)$, $c = (c_1, c_1, \dots, c_n)$ a matice $A = ((a_{1,1}, \dots, a_{1,n}), (a_{2,1}, \dots, a_{2,n}), \dots, (a_{n,1}, \dots, a_{n,n}))$ a hodnot $x_0, x_T, y_0 = (y_{0,1}, y_{0,2}, \dots, y_{0,n})$ a počet kroků n . A vypočte přibližnou hodnotu $y(x_T)$.

Algoritmus 2: Implicit Runge-Kutta

```
Result:  $yn$ 
 $h \leftarrow \frac{x_T - x_0}{n}$ 
 $yn \leftarrow y_0$ 
for  $i = 1..n$  do
     $xn \leftarrow x_0 + (i - 1)h$ 
    for  $j = 1..n$  do
         $g_j \leftarrow z_j - f(xn + c_j h, yn + h \sum_{k=1}^n a_{j,k} z_k)$ 
    end
     $Solve(g_1 = 0, \dots, g_n = 0) \text{ For } (z_1, \dots, z_n)$ 
     $yn \leftarrow yn + h \sum_{j=1}^n b_j z_j$ 
end
```

Algoritmus 3: SDIRK

```
Result:  $yn$ 
 $h \leftarrow \frac{x_T - x_0}{n}$ 
 $yn \leftarrow y_0$ 
for  $i = 1..n$  do
     $xn \leftarrow x_0 + (i - 1)h$ 
    for  $j = 1..n$  do
         $g \leftarrow z_j - f(xn + c_j h, yn + h \sum_{k=1}^j a_{j,k} z_k)$ 
         $Solve(g) \text{ For } (z_j)$ 
    end
     $yn \leftarrow yn + h \sum_{j=1}^n b_j z_j$ 
end
```

Dále, jak už bylo několikrát zmíněno, tato obecná verze (Algoritmus 2) je pro praktické účely typicky nevhodná, proto zde uvedeme ještě speciální verzi této metody, konkrétně SDIRK (singly-diagonal implicit Runge-Kutta), která bude mít stejné vstupy a výstupy jako předchozí algoritmus.

Na první pohled se může zdát, že tato metoda vyžaduje více práce, neboť musíme pro každý krok řešit n nelineárních soustav. Nově však soustavy mají n -krát méně proměnných, což je v případě řešení nelineárních soustav velká výhoda.

Pokud bychom opět chtěli získat výsledek s předem specifikovanou přesností, měli bychom se tentokrát vzhledem k ceně výpočtu vyhnout naivním metodám a rovnou využít nějakou sofistikovanější, jako příklad opět uvedeme adaptivní řízení velikosti kroku, které nalezneme v [8].

6.3 Galerkinova metoda

Galerkinova metoda je zcela odlišný přístup k numerickému řešení ODR oproti již zmiňovaným Runge-Kutta metodám. Např. rozdíl ve složitosti výpočtu bude značný pokud budeme řešit lineární, nebo nelineární úlohy. V teoretické části jsme si formulovali Galerkinovu metodu pouze pro lineární ODR, a proto začneme algoritmizací metody pro lineární úlohy. Metodu pro řešení nelineárních ODR také uvedeme, avšak nebude nijak teoreticky podložená, pouze empiricky testována.

V první části tedy budeme uvažovat diferenciální rovnici ve tvaru

$$Y'(x) + F(x)Y(x) = G(x), \quad Y(x_0) = Y_0 \quad (6.1)$$

kde

$$F(x) = \begin{bmatrix} f_{1,1}(x) & f_{1,2}(x) & \cdots & f_{1,n}(x) \\ f_{2,1}(x) & f_{2,2}(x) & \cdots & f_{2,n}(x) \\ \vdots & \vdots & \ddots & \vdots \\ f_{n,1}(x) & f_{n,2}(x) & \cdots & f_{n,n}(x) \end{bmatrix}$$

$$Y(x) = \begin{bmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_n(x) \end{bmatrix} \quad Y'(x) = \begin{bmatrix} y'_1(x) \\ y'_2(x) \\ \vdots \\ y'_n(x) \end{bmatrix} \quad G(x) = \begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_n(x) \end{bmatrix}$$

V teoretické části jsme aproximovali řešení ve vektorovém prostoru, to by však znamenalo, že $Y_0 = (0, \dots, 0)$. My však chceme řešit obecnější formu počáteční úlohy, proto budeme řešení hledat jako $\bar{Y} = Y_0 + \bar{Y}_h$, kde \bar{Y}_h je homogenní řešení.

Pro naše potřeby budeme všechny složky homogenního řešení aproximovat ve stejném vektorovém prostoru V_m s bází $\langle \phi_1(x), \dots, \phi_m(x) \rangle$, takže dostáváme

$$\bar{Y}_h(x) = \begin{bmatrix} \alpha_{1,1} \\ \alpha_{2,1} \\ \vdots \\ \alpha_{n,1} \end{bmatrix} \phi_1(x - x_0) + \begin{bmatrix} \alpha_{1,2} \\ \alpha_{2,2} \\ \vdots \\ \alpha_{n,2} \end{bmatrix} \phi_2(x - x_0) + \dots + \begin{bmatrix} \alpha_{1,m} \\ \alpha_{2,m} \\ \vdots \\ \alpha_{n,m} \end{bmatrix} \phi_m(x - x_0) \quad (6.2)$$

Dále je potřeba výraz (6.2) dosadit do rovnice (6.1) a obě strany nestandardně vynásobit vektorem bázových funkcí. Nejprve však ještě odečteme nehomogenitu $F(x)Y_0$ od pravé strany rovnice (6.1). Dostáváme levou L^* a pravou R^* stranu rovnice

$$\begin{aligned} L &= \bar{Y}'_h(x) + F(x)\bar{Y}_h(x) & R &= G(x) - F(x)Y_0 \\ L^* &= \begin{bmatrix} L_1 \cdot \phi_1 \\ \vdots \\ L_1 \cdot \phi_m \\ L_2 \cdot \phi_1 \\ \vdots \\ \vdots \\ L_n \cdot \phi_m \end{bmatrix} & R^* &= \begin{bmatrix} R_1 \cdot \phi_1 \\ \vdots \\ R_1 \cdot \phi_m \\ R_2 \cdot \phi_1 \\ \vdots \\ \vdots \\ R_n \cdot \phi_m \end{bmatrix} \end{aligned}$$

Neboť jsme vycházeli ze soustavy lineárních ODR, můžeme L^* vyjádřit jako součin matice s vektorem α .

Nakonec každý prvek této matice a vektoru R^* zintegrujeme podle x , dostáváme lineární systém, jehož vyřešením obdržíme koeficienty α

$$\begin{bmatrix} \ell_{1,1}^{1,1} & \dots & \ell_{1,1}^{m,1} & \widehat{\ell}_{1,2}^{1,1} & \dots & \dots & \widehat{\ell}_{1,n}^{m,1} \\ \vdots & & \vdots & \vdots & & & \vdots \\ \ell_{1,1}^{1,m} & \dots & \ell_{1,1}^{m,m} & \widehat{\ell}_{1,2}^{1,m} & \dots & \dots & \widehat{\ell}_{1,n}^{m,m} \\ \widehat{\ell}_{2,1}^{1,1} & \dots & \widehat{\ell}_{2,1}^{m,1} & \ell_{2,2}^{1,1} & \dots & \dots & \widehat{\ell}_{2,n}^{m,1} \\ \vdots & & \vdots & \vdots & & & \vdots \\ \vdots & & \vdots & \vdots & & & \vdots \\ \widehat{\ell}_{n,1}^{1,m} & \dots & \widehat{\ell}_{n,1}^{m,m} & \widehat{\ell}_{n,2}^{1,m} & \dots & \dots & \ell_{n,n}^{m,m} \end{bmatrix} \begin{bmatrix} \alpha_{1,1} \\ \vdots \\ \alpha_{1,m} \\ \alpha_{2,1} \\ \vdots \\ \vdots \\ \alpha_{n,m} \end{bmatrix} = \begin{bmatrix} r_1^1 \\ \vdots \\ r_1^m \\ r_2^1 \\ \vdots \\ \vdots \\ r_n^m \end{bmatrix} \quad (6.3)$$

kde

$$\begin{aligned} \ell_{a,b}^{c,d} &= \int_{x_0}^{x_T} (\phi'_c(x) + f_{a,b}(x)\phi_c(x))\phi_d(x) dx \\ \widehat{\ell}_{a,b}^{c,d} &= \int_{x_0}^{x_T} f_{a,b}(x)\phi_c(x)\phi_d(x) dx \\ r_a^b &= \int_{x_0}^{x_T} \left(g_a(x) - \sum_{i=1}^n f_{a,i}(x)y_{0,i} \right) \phi_b(x) dx \end{aligned}$$

Symbolický algoritmus tentokrát uvádět nebudeme, jedná se totiž pouze o sestavení a vyřešení soustavy (6.3) podle návodu v této sekci. Co se týče samotného řešení soustavy (6.3), jedná se většinou o velmi špatně podmíněný systém, což má za následek nestabilitu metody pro velká m . V našem případě použijeme k řešení SVD rozklad, nestabilitu tak zmírníme, ale neodstraníme.

Dále si popíšeme použití této metody na nelineární soustavy ODR, jak už bylo zmíněno na začátku sekce, tuto metodu nemáme nijak teoreticky odvozenou. Budeme tedy následovat „logickou“ cestu od řešiče lineárních systémů k nelineárním. Uvažujme tedy soustavu ODR

$$Y'(x) = F(x, Y(x)), \quad Y(x_0) = Y_0 \quad (6.4)$$

kde

$$Y(x) = \begin{bmatrix} y_1(x) \\ y_2(x) \\ \vdots \\ y_n(x) \end{bmatrix} \quad Y'(x) = \begin{bmatrix} y'_1(x) \\ y'_2(x) \\ \vdots \\ y'_n(x) \end{bmatrix} \quad F(x) = \begin{bmatrix} f_1(x, y_1(x), \dots, y_n(x)) \\ f_2(x, y_1(x), \dots, y_n(x)) \\ \vdots \\ f_n(x, y_1(x), \dots, y_n(x)) \end{bmatrix}$$

Obdobně jako u lineární metody dosadíme \bar{Y} do (6.4), vše převedeme na levou stranu rovnice a opět provedeme nestandardní součin s vektorem ϕ

$$L = \bar{Y}'(x) - F(x, \bar{Y}(x))$$

$$L^* = \begin{bmatrix} L_1 \cdot \phi_1 \\ \vdots \\ L_1 \cdot \phi_m \\ L_2 \cdot \phi_1 \\ \vdots \\ \vdots \\ L_n \cdot \phi_m \end{bmatrix} \quad R^* = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

Na rozdíl od lineární metody tentokrát nic nevytýkáme. Jsme nuceni řešit soustavu $n \cdot m$ nelineárních rovnic o $n \cdot m$ neznámých α

$$L^*(\alpha) = R^*$$

$$\begin{bmatrix} \int_{x_0}^{x_T} [\bar{y}'_1(x) - f_1(x, \bar{y}_1(x), \dots, \bar{y}_n(x))] \phi_1(x) dx \\ \vdots \\ \int_{x_0}^{x_T} [\bar{y}'_1(x) - f_1(x, \bar{y}_1(x), \dots, \bar{y}_n(x))] \phi_m(x) dx \\ \int_{x_0}^{x_T} [\bar{y}'_2(x) - f_2(x, \bar{y}_1(x), \dots, \bar{y}_n(x))] \phi_1(x) dx \\ \vdots \\ \vdots \\ \int_{x_0}^{x_T} [\bar{y}'_n(x) - f_n(x, \bar{y}_1(x), \dots, \bar{y}_n(x))] \phi_m(x) dx \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

Samotný algoritmus opět neuvedeme, je však potřeba zmínit několik detailů. Podobně jako u implicitní Runge-Kutty rychlost této metody přímo závisí na rychlosti řešení nelineárních soustav. V praxi je tedy potřeba zvážit, zda se vůbec vyplatí ji použít na daný problem, v mnoha případech je vhodnější použít explicitní Runge-Kutta metodu s malým krokem pro dosažení srovnatelných, nebo lepších výsledků.

Galerkinova metoda na rozdíl od metod Runge-Kutta vrací koeficienty α , pokud tedy chceme získat konkrétní hodnoty aproximace Y v nějakém bodě $x \in \langle x_0, x_T \rangle$, musíme dosadit zpět do (6.2) a přičíst Y_0 . Tento tvar řešení také přináší několik výhod. Protože si báze funkce ϕ volí uživatel (tudíž je zná předem), můžeme po numerickém nalezení koeficientů α použít nástroje matematické analýzy k hledání extrémů, nebo inflexních bodů.

6.4 Kolokační metoda

Spektrální Galerkinova metoda dosahuje exponenciální konvergence pouze v případě, kdy je řešení diferenciální rovnice analytická funkce. Pokud toto vezmeme v potaz a navíc přidáme fakt, že numerický výpočet integrálů v Galerkinově metodě bude prováděn jako vážený součet integrandů vyhodnocených v diskrétních bodech, můžeme formulovat tzv. kolokační metodu. Tuto metodu nemáme nijak teoreticky podloženou, a tudíž její efektivitu budeme pouze empiricky testovat na modelových problémech.

Co se samotného výpočtu týče, je kolokační metoda téměř stejně jako metoda Galerkinova, proto nebudeme rozepisovat detaily výpočtu jako v předchozí sekci. Začneme tedy metodou pro řešení lineárních systémů ve tvaru (6.1), kde hledané řešení bude mít opět tvar $\bar{Y} = Y_0 + \bar{Y}_h$, kde \bar{Y}_h má tvar (6.2). Dostáváme se k prvnímu rozdílu mezi kolokační a Galerkinovou metodou a to konkrétně k vyjádření L^* a R^* , místo násobení jednotlivých složek L a R báze funkcemi ϕ a integrování, budeme tyto složky evaluovat v bodech (x_1, \dots, x_m) takto

$$L = \bar{Y}'_h(x) + F(x)\bar{Y}_h(x) \qquad R = G(x) - F(x)Y_0$$

$$L^* = \begin{bmatrix} L_1(x_1) \\ \vdots \\ L_1(x_m) \\ L_2(x_1) \\ \vdots \\ \vdots \\ L_n(x_m) \end{bmatrix} \qquad R^* = \begin{bmatrix} R_1(x_1) \\ \vdots \\ R_1(x_m) \\ R_2(x_1) \\ \vdots \\ \vdots \\ R_n(x_m) \end{bmatrix}$$

Díky linearitě původní soustavy můžeme opět zapsat L^* jako součin matice s vektorem α . Výsledná lineární soustava tedy dostává tvar

$$\begin{bmatrix} \ell_{1,1}^{1,1} & \dots & \ell_{1,1}^{m,1} & \widehat{\ell}_{1,2}^{1,1} & \dots & \dots & \widehat{\ell}_{1,n}^{m,1} \\ \vdots & & \vdots & \vdots & & & \vdots \\ \ell_{1,1}^{1,m} & \dots & \ell_{1,1}^{m,m} & \widehat{\ell}_{1,2}^{1,m} & \dots & \dots & \widehat{\ell}_{1,n}^{m,m} \\ \widehat{\ell}_{2,1}^{1,1} & \dots & \widehat{\ell}_{2,1}^{m,1} & \ell_{2,2}^{1,1} & \dots & \dots & \widehat{\ell}_{2,n}^{m,1} \\ \vdots & & \vdots & \vdots & & & \vdots \\ \vdots & & \vdots & \vdots & & & \vdots \\ \widehat{\ell}_{n,1}^{1,m} & \dots & \widehat{\ell}_{n,1}^{m,m} & \widehat{\ell}_{n,2}^{1,m} & \dots & \dots & \ell_{n,n}^{m,m} \end{bmatrix} \begin{bmatrix} \alpha_{1,1} \\ \vdots \\ \alpha_{1,m} \\ \alpha_{2,1} \\ \vdots \\ \vdots \\ \alpha_{n,m} \end{bmatrix} = \begin{bmatrix} r_1^1 \\ \vdots \\ r_1^m \\ r_2^1 \\ \vdots \\ \vdots \\ r_n^m \end{bmatrix} \quad (6.5)$$

kde

$$\begin{aligned} \ell_{a,b}^{c,d} &= \phi'_c(x_d) + f_{a,b}(x_d)\phi_c(x_d) \\ \widehat{\ell}_{a,b}^{c,d} &= f_{a,b}(x_d)\phi_c(x_d) \\ r_a^b &= g_a(x_d) - \sum_{i=1}^n f_{a,i}(x_d)y_{0,i} \\ x_i &\in \langle x_0, x_T \rangle, \quad i \in \{1, \dots, m\} \end{aligned}$$

Vyřešením této soustavy obdržíme požadované váhy bázeových funkcí.

Analogicky v případě řešení nelineárních soustav ODR bude potřeba vyřešit následující soustavu nelineárních rovnic pro neznámé α

$$\begin{bmatrix} \overline{y}'_1(x_1) - f_1(x_1, \overline{y}_1(x_1), \dots, \overline{y}_n(x_1)) \\ \vdots \\ \overline{y}'_1(x_m) - f_1(x_m, \overline{y}_1(x_m), \dots, \overline{y}_n(x_m)) \\ \overline{y}'_2(x_1) - f_2(x_1, \overline{y}_1(x_1), \dots, \overline{y}_n(x_1)) \\ \vdots \\ \vdots \\ \overline{y}'_n(x_m) - f_n(x_m, \overline{y}_1(x_m), \dots, \overline{y}_n(x_m)) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

Z hlediska složitosti a podmíněnosti jak lineárních, tak nelineárních soustav tato metoda oproti Galerkinově metodě nic nepřináší. Co se však samotného výpočtu týče lze argumentovat, že je rychlejší, např. nemusíme numericky řešit velké množství integrálů, což má za následek méně evaluací funkcí F a G . Tento fakt je pak ještě znatelnější v případě řešení nelineárních ODR. Rozmístění jednotlivých $x_i \in \langle x_0, x_T \rangle$, $i \in \{1, \dots, m\}$ může být obecně libovolné dokud $x_i \neq x_j$, $i \neq j$. Empirické výsledky však ukázaly, že volba x_i jako kořenů Legendreova, nebo Chebyshevova polynomu dává lepší výsledky z hlediska chyby aproximace, než jejich volba náhodná, nebo ekvidistantní. Vlastnosti výsledného řešení jsou srovnatelné s Galerkinovou metodou.

7. Numerické experimenty

V předchozích kapitolách jsme uvedli několik metod jak z hlediska teorie, tak jejich algoritmického provedení. Jelikož se jedná o numerické metody, jejichž primární rolí je řešit praktické úlohy, je extrémně důležité tyto metody řádně otestovat a porovnat mezi sebou právě na praktických příkladech. Pokud nebude řečeno jinak, všechny numerické experimenty budou replikovatelné pomocí skriptů v elektronické příloze. Skripty jsou napsány v Octave a jsou kompatibilní s Matlabem. Skript s jednotlivými experimenty se jmenuje „method_testing.m“, v němž každý experiment má unikátní „ID“, které se bude shodovat s ID uvedeným v této práci, pokud nebude řečeno jinak. Pro zjednodušení budeme v této kapitole označovat n stavovou Runge-Kutta metodu s m -tým řádem konvergence $ERK_n(m)$ v případě explicitní a $IRK_n(m)$ v případě implicitní verze.

7.1 Ukázka jednotlivých metod

1. Řešení lineární ODR (ID 1)

Začneme nejprve ukázkou řešení lineární ODR se známým řešením pomocí metod $ERK_2(2)$, $IRK_1(2)$, Galerkinové a kolokační s monomiály jakožto bázovými funkcemi.

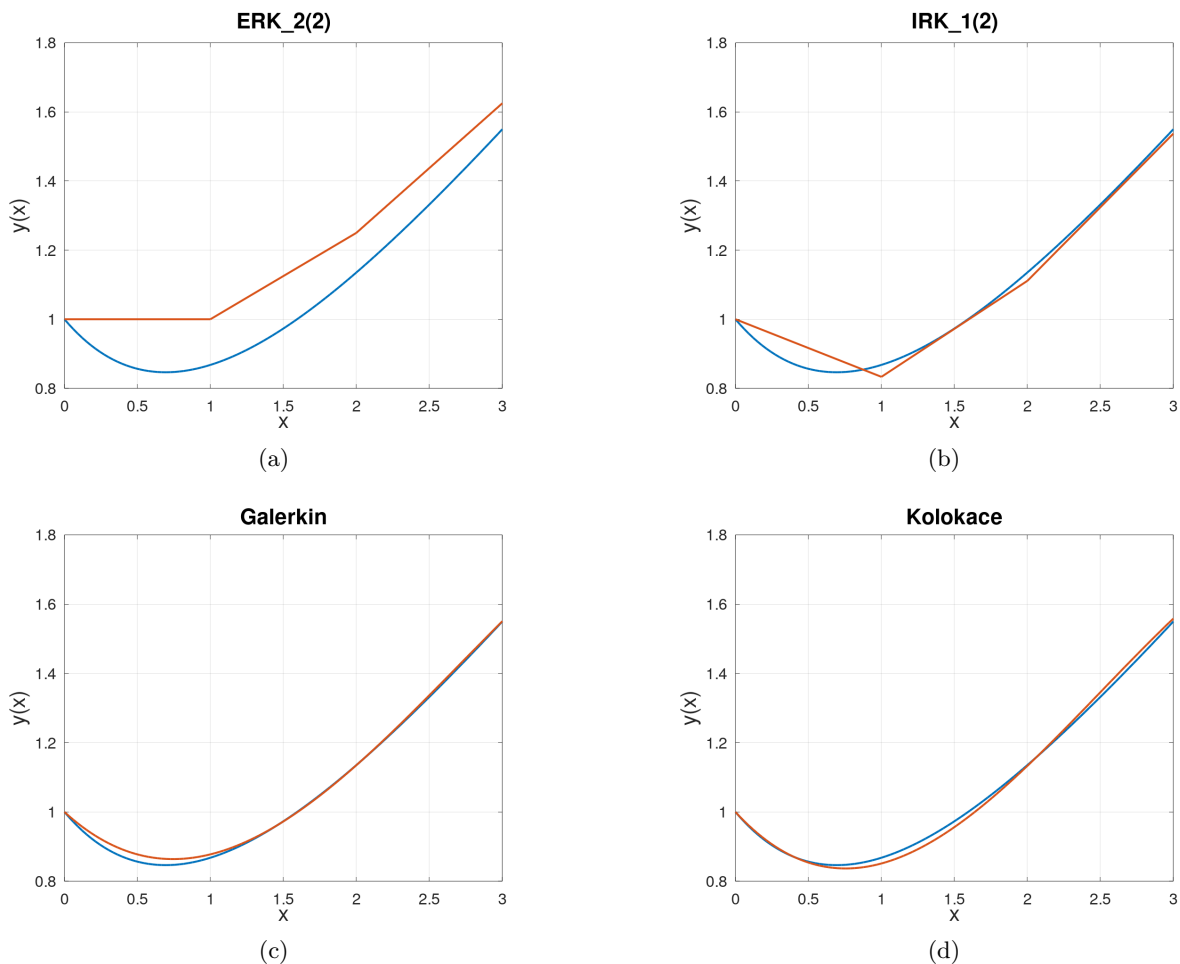
Řešená ODR bude

$$y'(x) + y(x) = \frac{x+1}{2}, \quad y(0) = 1, \quad x_T = 3$$

s analytickým řešením

$$y(x) = e^{-x} + \frac{x}{2}$$

Na obrázku 7.1 můžeme vidět mírnou nestabilitu v případě použití explicitní Runge-Kutty (a), která zmizí v případě použití Runge-Kutty implicitní (b), důvod je lépe popsán v kapitole 4. Co se týče Galerkinovy (c) a kolokační (d) metody máme relativně přesné řešení, i když je n pouze 3 (při výpočtu byly použity pouze 3 bázové funkce, konkrétně x , x^2 a x^3).



Obrázek 7.1: Ukázka řešení lineární ODR pomocí některých metod, přibližné řešení modře, analytické řešení oranžově, $n = 3$

2. Řešení nelineární ODR (ID 2)

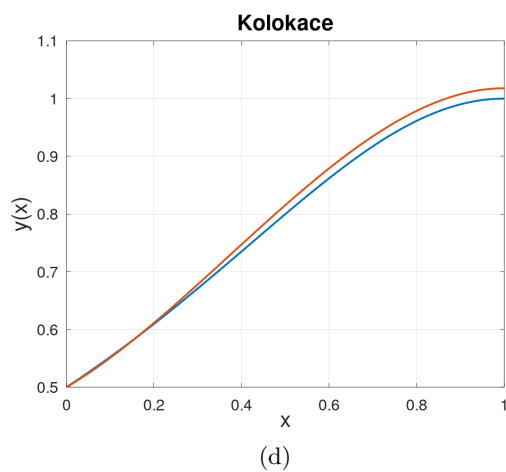
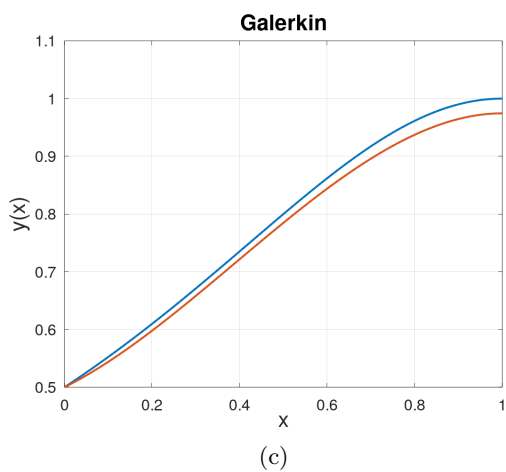
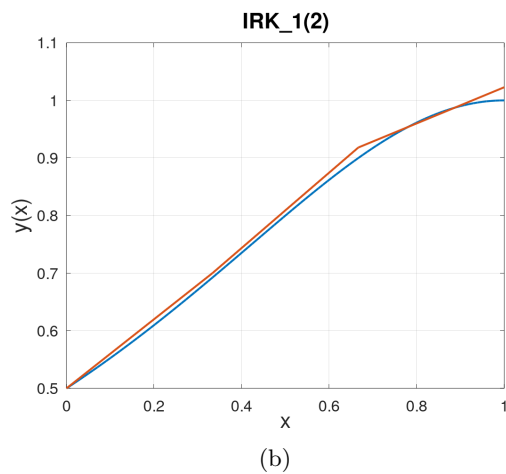
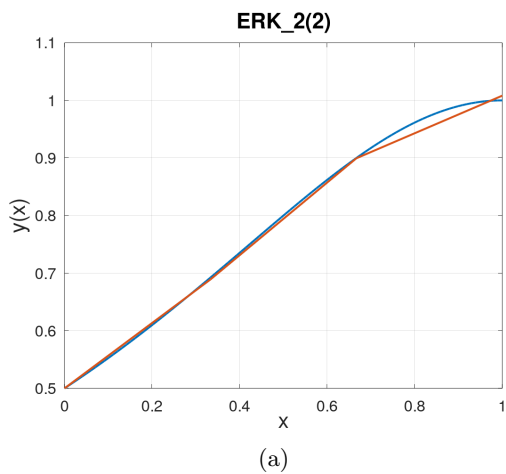
Dále si ukážeme příklad řešení nelineární ODR se známým řešením pomocí metod $ERK_2(2)$, $IRK_1(2)$, Galerkinové a kolokační s monomiály jakožto bázovými funkcemi.

Řešená ODR bude

$$y'(x) = (2 - 2x)y^2(x), \quad y(0) = \frac{1}{2}, \quad x_T = 1$$

s analytickým řešením

$$y(x) = \frac{1}{1 + (x - 1)^2}$$



Obrázek 7.2: Ukázka řešení nelineární ODR pomocí některých metod, přibližné řešení modře, analytické řešení oranžově, $n = 3$

Obrázek 7.2 neukazuje žádný velký rozdíl mezi jednotlivými metodami, avšak je potřeba zmínit, i když to nebyl předmět testování, trvání výpočtu v případě Galerkinovy metody bylo oproti metodám Runge-Kutta značné.

7.2 Ověření řádů konvergence

1. Srovnání řádů konvergence Runge-Kutta metod (ID 3)

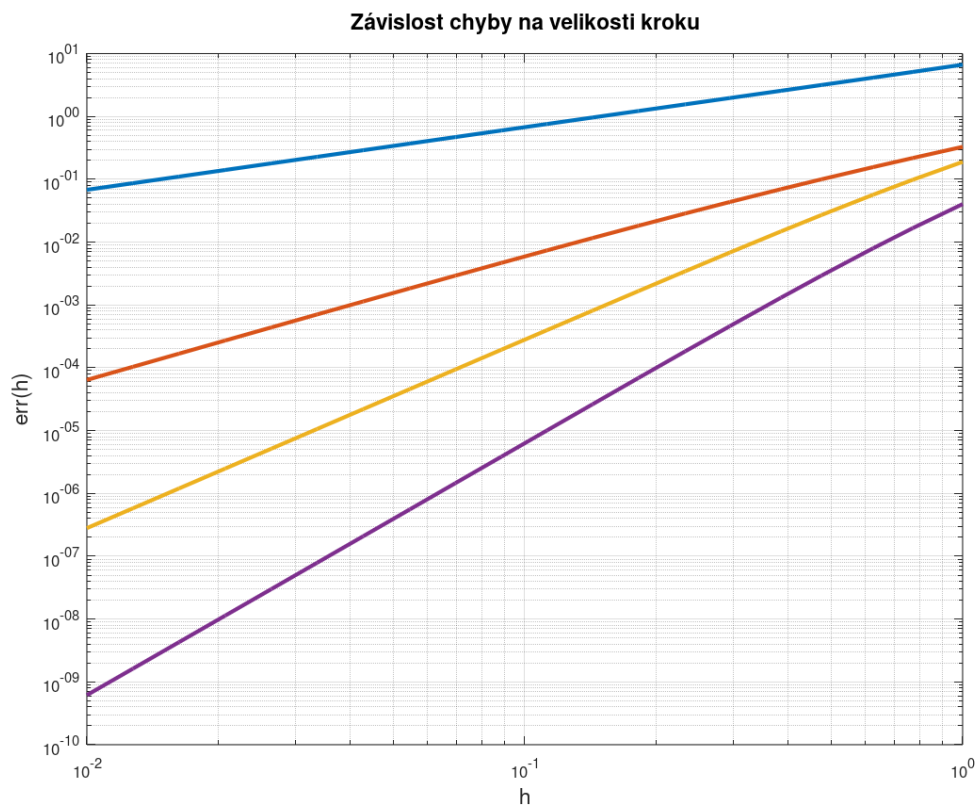
Zde srovnáme závislosti chyb explicitních Runge-Kutta metod na velikosti kroku. Chybou bude myšlena vzdálenost mezi analytickým a přibližným řešením v bodě x_T . Srovnávat budeme metody $ERK_1(1)$, $ERK_2(2)$, $ERK_3(3)$ a $ERK_4(4)$.

Řešená ODR bude

$$y'(x) - \frac{y(x)}{x+1} = (x+1)\cos(x), \quad y(0) = 0, \quad x_T = 10$$

s analytickým řešením

$$y(x) = (x+1)\sin(x)$$



Obrázek 7.3: Srovnání rychlostí konvergence $ERK_1(1)$ modře, $ERK_2(2)$ oranžově, $ERK_3(3)$ žlutě a $ERK_4(4)$ fialově

Obrázek 7.3 ukazuje závislosti chyb na velikosti kroku v logaritmické měřítku. Jelikož metody Runge-Kutta n -tého řádu mají globální chybu řádu $err = \mathcal{O}(h^n)$, v logaritmickém měřítku nám tato rovnost dá $\log(err) = n\mathcal{O}(\log(h))$. V logaritmickém měřítku by tedy průměrný sklon jednotlivých závislostí chyby na velikosti kroku měl odpovídat řádu konvergence metody, což se potvrdilo.

2. Srovnání řádů konvergence Runge-Kutta, Galerkinovy a kolokační metody (ID 4)

Jedná se o obdobné srovnání, jako v případě Runge-Kutta metod. Jediný rozdíl je v tom, co myslíme chybou, tentokrát ji budeme měřit jako

$$err_n = \frac{1}{n} \sqrt{\sum_{i=0}^n \left(y\left(x_0 + \frac{i}{n}(x_T - x_0)\right) - y_n\left(x_0 + \frac{i}{n}(x_T - x_0)\right) \right)^2}$$

kde $y(x)$ je řešení analytické a $y_n(x)$ je řešení přibližné. Tuto chybu jsme zvolili záměrně tak, aby metoda Runge-Kutta nebyla proti metodě Galerkinové/kolokační znevýhodněna. Řešit budeme pomocí metod $ERK_4(4)$, Galerkin a kolokace s Chebyshevovými polynomy jakožto базовými funkcemi.

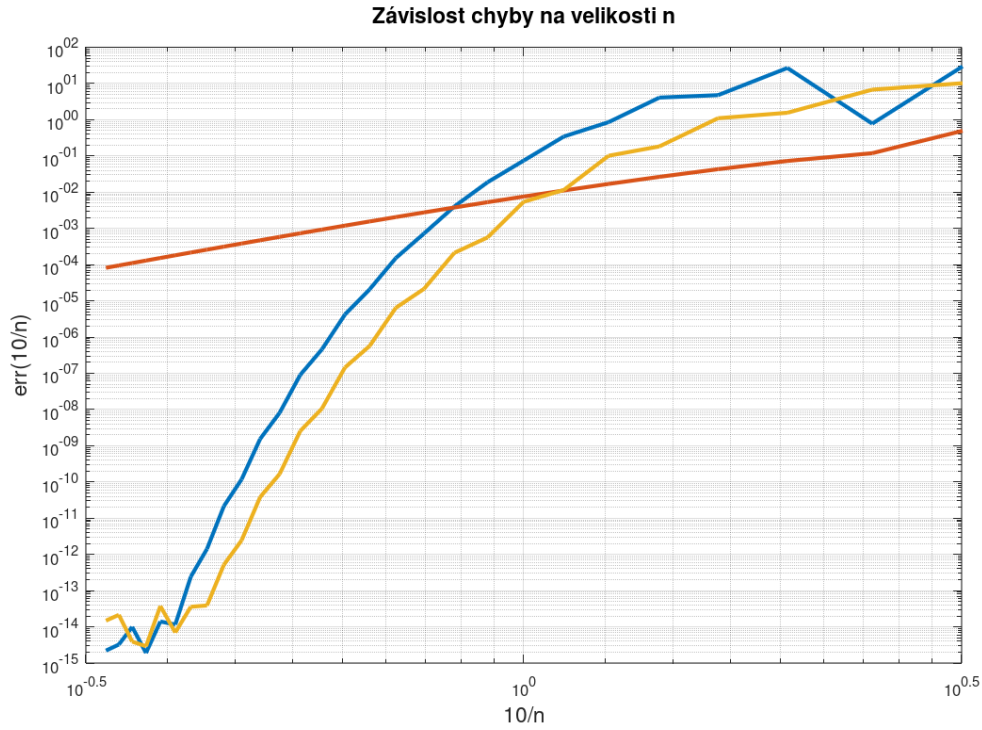
Řešená ODR bude

$$y'(x) - \frac{y(x)}{x+1} = (x+1)\cos(x), \quad y(0) = 0, \quad x_T = 10$$

s analytickým řešením

$$y(x) = (x+1)\sin(x)$$

Obrázek 7.4 obdobně jako obrázek 7.3 ukazuje závislosti chyb na velikosti kroku v logaritmické měřítku. Můžeme tak, z dříve uvedených důvodů, vidět dominantní exponenciální konvergenci Galerkinové i kolokační metody. Pro tuto rovnici dokonce kolokace dává přesnější výsledky. Samozřejmě metody kolokace i Galerkin přestanou dále konvergovat po dosažení maximální přesnosti výpočtu s plovoucí desetinnou čárkou (v tomto případě datového typu double).



Obrázek 7.4: Srovnání rychlostí konvergence $ERK_4(4)$ oranžově, kolokační metody žlutě a Galerkinovy metody modře

3. Srovnání řádů konvergence Runge-Kutta, Galerkinovy a kolokační metody pro ODR s neanalytickým řešením (ID 6)

Zde si ukážeme účinnost metod při řešení problému, jehož řešení není analytické. Použijeme stejný výpočet chyby jako v předchozím srovnání. Řešit budeme pomocí metod $ERK_4(4)$, Galerkin a kolokace s Chebyshevovými polynomy jakožto bázovými funkcemi.

Řešená ODR bude

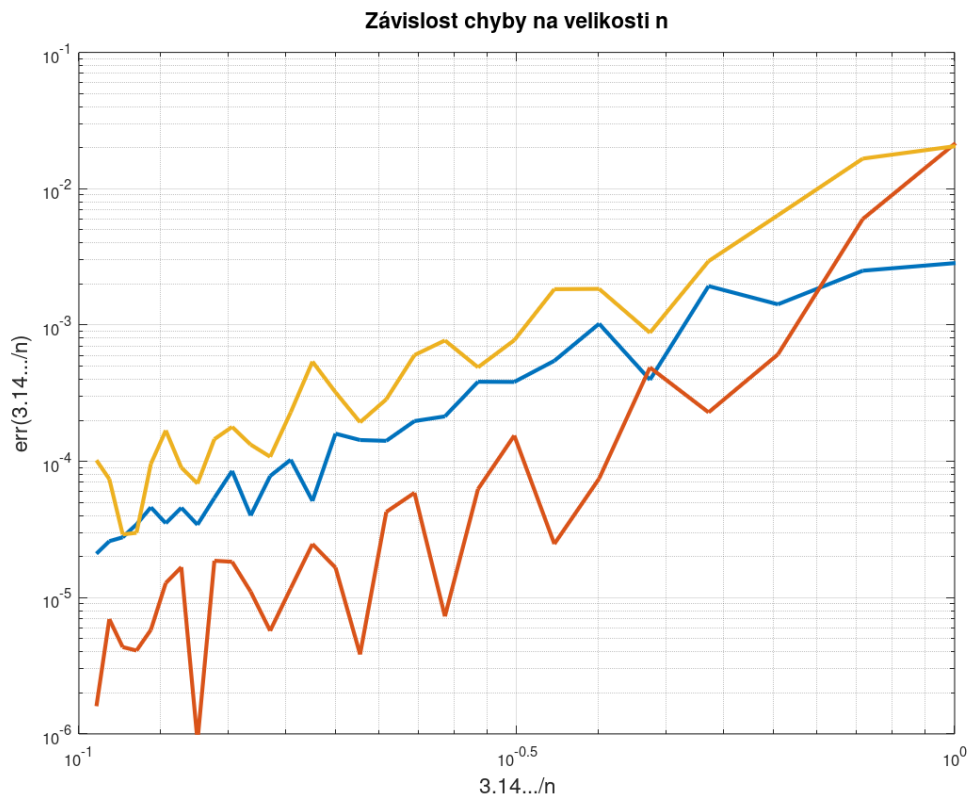
$$y'(x) + y(x) = (x-1)\theta(x-1), \quad y(0) = 1, \quad x_T = \pi$$

s analytickým řešením

$$y(x) = e^{-x} + (e^{1-x} + x - 2)\theta(x-1)$$

kde

$$\theta(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}$$



Obrázek 7.5: Srovnání rychlostí konvergence $ERK_4(4)$ oranžově, kolokační metody žlutě a Galerkinovy metody modře

V experimentu na obrázku 7.5 můžeme vidět spektrální konvergenci. Protože řešení oné rovnice náleží $H^2((0, x_T))$, tak očekáváme konvergenci 2 řádu, což je z obrázku dobře vidět. Dále vidíme také pokles rychlosti konvergence Runge-Kutta metody, to je zapříčiněno nespojitostí vyšších derivací řešení v bodě $x = 1$, tudíž v tomto bodě neexistuje Taylorův polynom dostatečně vysokého řádu, který Runge-Kutta využívá k výpočtu následujícího kroku. Ještě je potřeba dodat, že v tomto případě Galerkinova ani kolokační metoda nepřináší oproti metodám Runge-Kutta žádný benefit.

7.3 Řešení soustav ODR

1. Řešení tuhých soustav ODR (ID 8)

O tuhých diferenciálních rovnicích jsme se zmínili již v kapitole 4, teď je potřeba otestovat jednotlivé metody k jejich řešení. Řešit budeme pomocí metod $ERK_4(4)$, $IRK_3(3)$, Galerkin a kolokace s Chebyshevovými polynomy jakožto базовými funkcemi.

Řešená soustava ODR bude

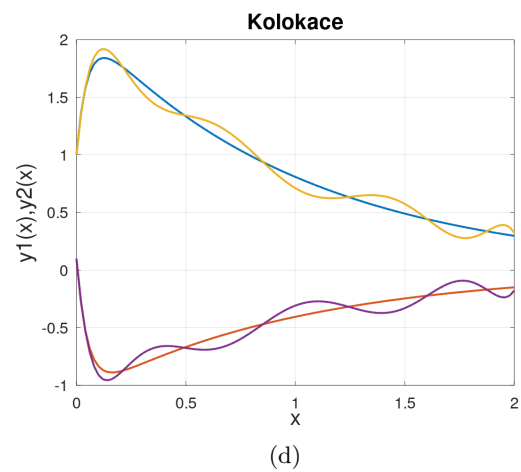
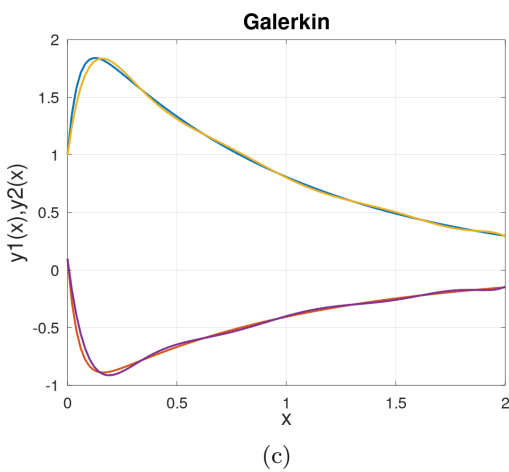
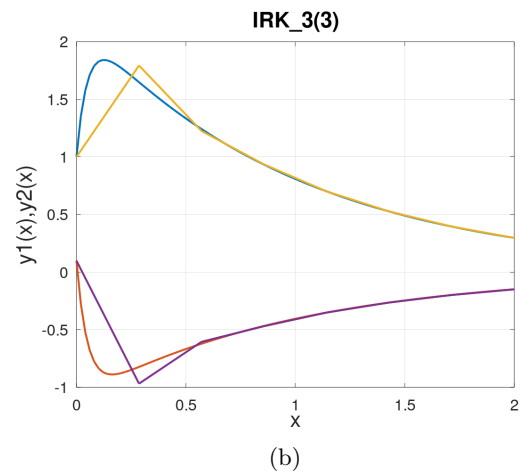
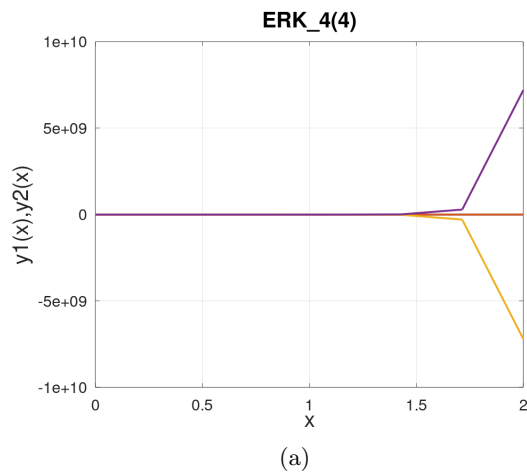
$$y_1'(x) - 18y_1(x) - 38y_2(x) = 0, \quad y_1(0) = 1$$

$$y_2'(x) + 19y_1(x) + 39y_2(x) = 0, \quad y_2(0) = \frac{1}{10}$$

s analytickým řešením

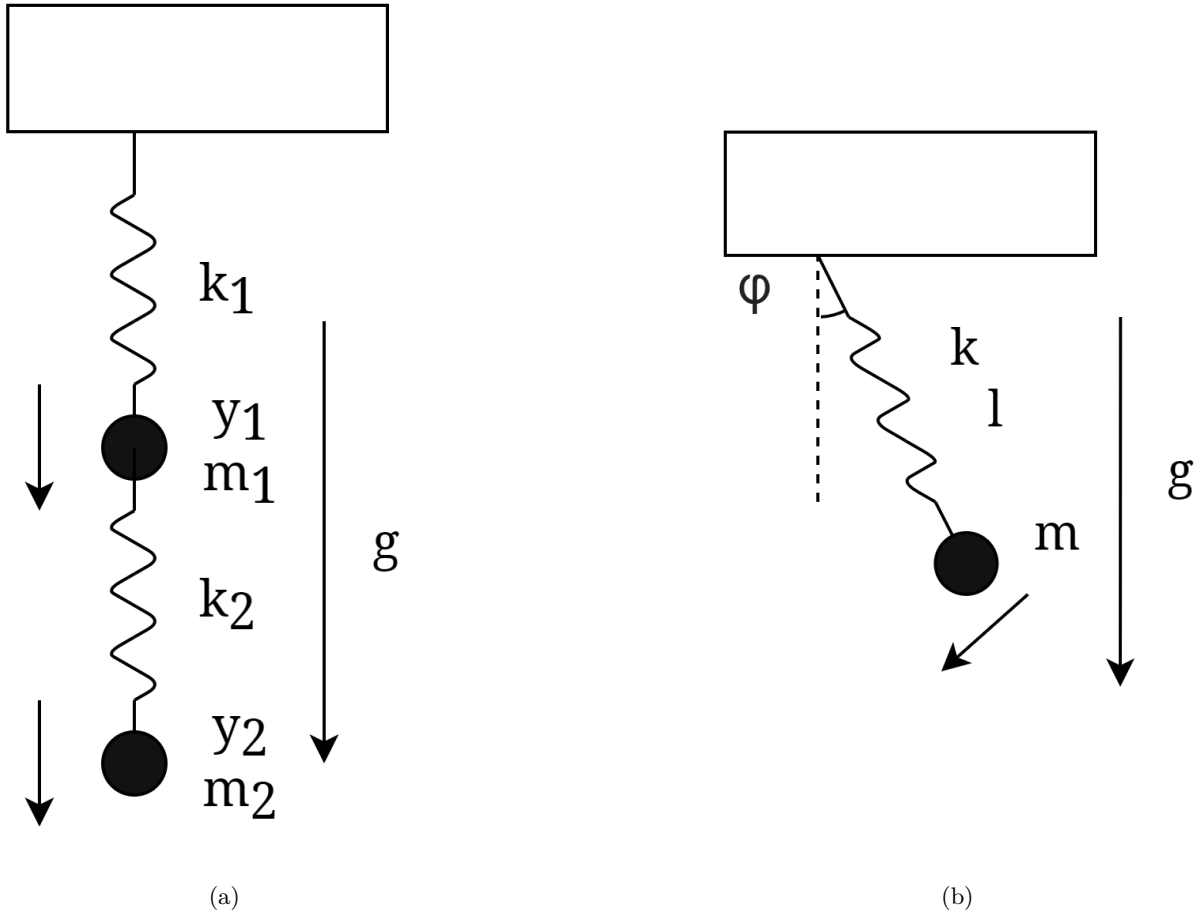
$$y_1(x) = 2, 2e^{-x} - 1, 2e^{-20x}$$

$$y_2(x) = 1, 2e^{-20x} - 1, 1e^{-x}$$



Obrázek 7.6: Řešení tuhých soustav ODR, přibližné řešení y_1 žlutě a y_2 fialově, analytické řešení y_1 modře a y_2 oranžově, $n = 8$

Na obrázku 7.6 vidíme extrémní nestabilitu v případě použití explicitní metody Runge-Kutty (a), která opět zmizí v případě použití metody Runge-Kutty implicitní (b), důvod je lépe popsán v kapitole 4. Co se týče Galerkinovy (c) a kolokační (d) metody dostáváme stabilní řešení, i když v případě Galerkina je řešení výrazně přesnější. Jelikož je řešení analytické, Galerkinova i kolokační metoda bude konvergovat exponenciálně.



Obrázek 7.7: Diagramy dvoupružinového systému (a) a kyvadla na pružině (b)

2. Řešení modelové lineární úlohy (ID 9)

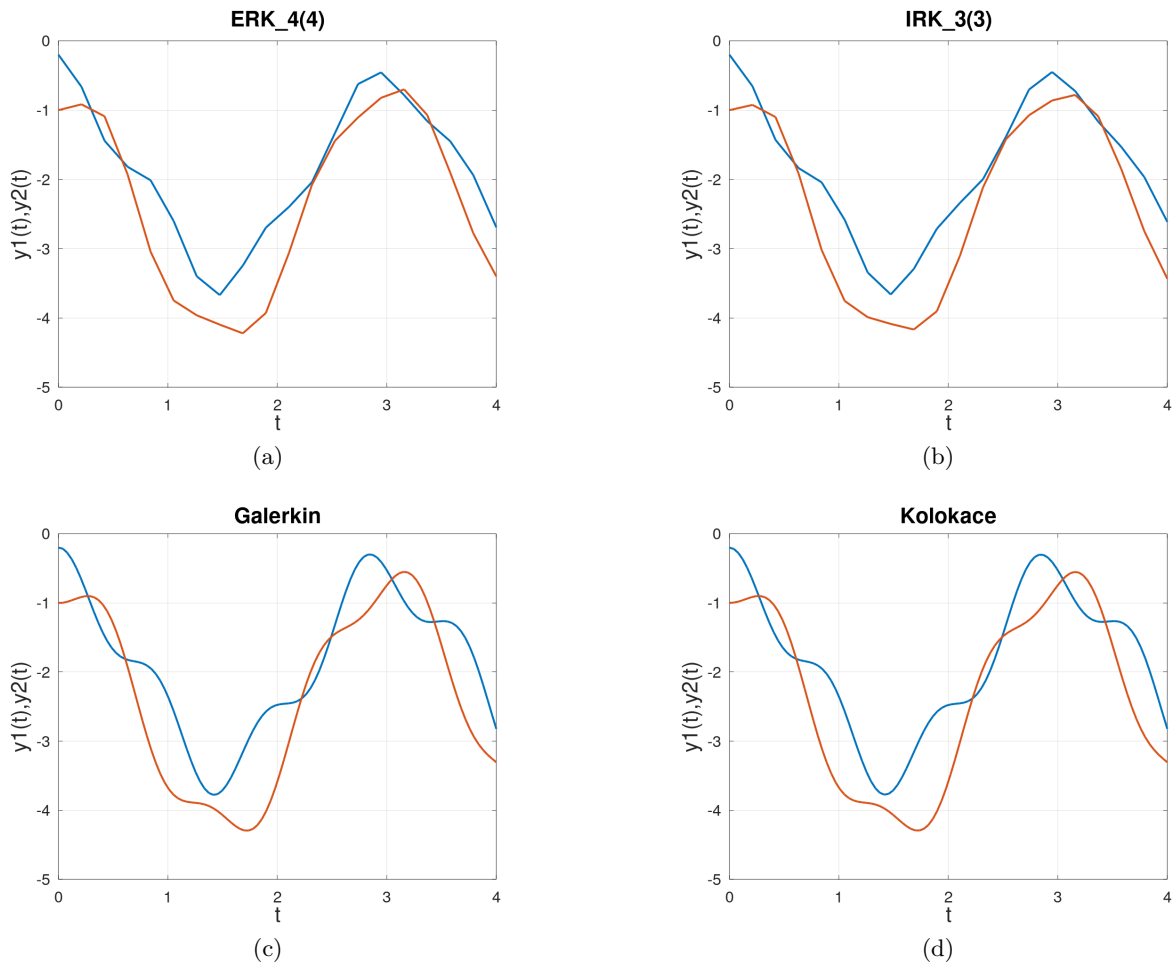
Zde se zaměříme na řešení lineárního fyzikálního systému, konkrétně dvoupružinové soustavy (obrázek 7.7a). Nejprve odvodíme soustavu diferenciálních rovnic z klasické mechaniky

$$\begin{aligned} m_1 y_1''(t) + k_1 y_1(t) - k_2 (y_2(t) - y_1(t)) + g &= 0 \\ m_2 y_2''(t) + k_2 (y_2(t) - y_1(t)) + g &= 0 \end{aligned}$$

Jedná se o soustavu dvou ODR druhého řádu, tudíž provedeme substituci tak, abychom dostali soustavu řádu prvního. Zároveň zvolíme $m_1 = m_2 = 1$, $k_1 = 10$, $k_2 = 20$ a $g = 10$. Dostáváme

$$\begin{aligned}y_1'(t) &= y_3(t) \\y_2'(t) &= y_4(t) \\y_3'(t) &= -10y_1(t) + 20(y_2(t) - y_1(t)) - 10 \\y_4'(t) &= -20(y_2(t) - y_1(t)) - 10\end{aligned}$$

Ještě zvolme počáteční podmínky $y_1(0) = -\frac{1}{5}$, $y_2(0) = -1$, $y_3(0) = 0$, $y_4(0) = 0$ a cíl $t_T = 4$. K řešení použijeme $ERK_4(4)$, $IRK_3(3)$, Galerkinu a kolokaci s Chebyshevovými polynomy jakožto bázovými funkcemi.



Obrázek 7.8: Řešení lineárního dvoupružinového systému pomocí několika metod, y_1 modře, y_2 oranžově, $n = 20$

Tento experiment 7.8 ukazuje, že všechny metody v této práci lze použít i k řešení lineárních soustav ODR vyšších řádů. Na tomto problému lze vidět, že Galerkinova (c) a kolokační (d) metoda dávají daleko přesnější řešení než obě Runge-Kutta metody (a) a (b).

3. Řešení modelové nelineární úlohy (ID 10)

Nakonec si ukážeme řešení nelineárního fyzikálního systému, konkrétně kyvadla na pružině (obrázek 7.7b). Nejprve odvodíme soustavu diferenciálních rovnic pomocí Lagrangeovy mechaniky

$$E_K = \frac{m}{2}(l'(t)^2 + l^2(t)\varphi'^2(t))$$

$$E_P = m g l(t) \cos(\varphi(t)) + \frac{1}{2}k l^2(t)$$

Dále využijeme Lagrangeovu rovnici

$$\frac{d}{dt} \left(\frac{\partial L}{\partial y'_i} \right) - \frac{\partial L}{\partial y_i} = Q_i, \quad L = E_K - E_P$$

V našem případě $i \in \{1, 2\}$, $Q_i = 0$ a y_i je $l(t)$ a $\varphi(t)$ respektive. Dosazením a zderivováním dostáváme

$$m l''(t) - (m l(t) \varphi'^2(t) - m g \cos(\varphi(t)) - k l(t)) = 0$$

$$2 m l(t) l'(t) \varphi'(t) + m l^2(t) \varphi''(t) - m g l(t) \sin(\varphi(t)) = 0$$

Vyjádříme $l''(t)$ a $\varphi''(t)$

$$l''(t) = l(t) \varphi'^2(t) - g \cos(\varphi(t)) - \frac{k}{m} l(t)$$

$$\varphi''(t) = \frac{-2 l'(t) \varphi'(t)}{l(t)} + \frac{g \sin(\varphi(t))}{l(t)}$$

Jedná se o soustavu dvou ODR druhého řádu, tudíž provedeme substituci tak, abychom dostali soustavu řádu prvního. Zároveň zvolíme $y_1(t) = l(t)$, $y_2(t) = \varphi(t)$, $m = 1$, $k = 10$ a $g = 10$. Dostáváme

$$y'_1(t) = y_3(t)$$

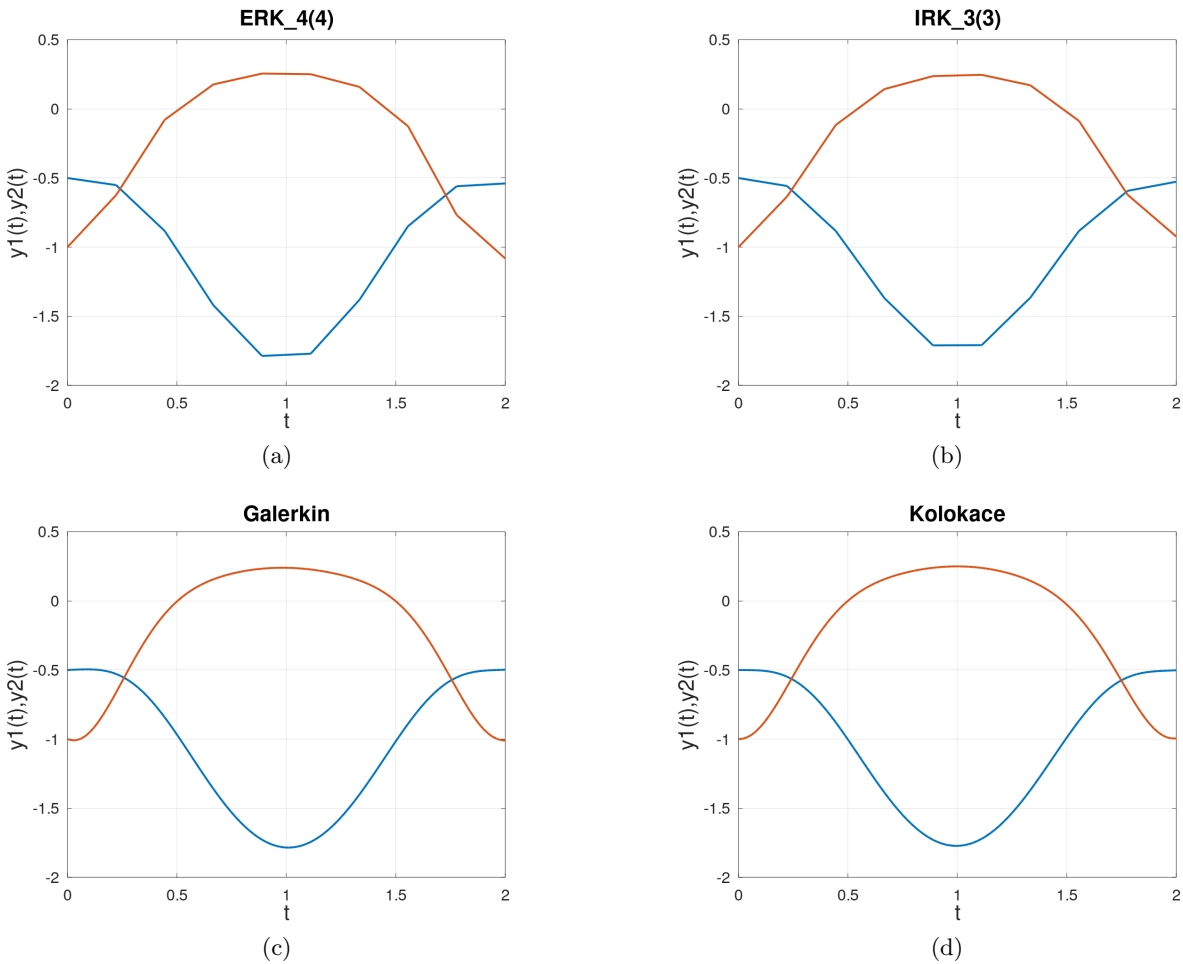
$$y'_2(t) = y_4(t)$$

$$y'_3(t) = y_1(t) y_4^2(t) - 10 \cos(y_2(t)) - 10 y_1(t)$$

$$y'_4(t) = \frac{-2 y_3(t) y_4(t)}{y_1(t)} + \frac{10 \sin(y_2(t))}{y_1(t)}$$

ještě zvolme počáteční podmínky $y_1(0) = -\frac{1}{2}$, $y_2(0) = -1$, $y_3(0) = 0$, $y_4(0) = 0$ a cíl $x_T = 2$.

K řešení použijeme $ERK_4(4)$, $IRK_3(3)$, Galerkina a kolokaci s Chebyshevovými polynomy jakožto bázovými funkcemi.



Obrázek 7.9: Řešení nelineárního kyvadla na pružině pomocí několika metod, y_1 modře, y_2 oranžově, $n = 10$

Poslední experiment (obrázek 7.9) ukazuje, že všechny metody v této práci lze použít i k řešení nelineárních soustav ODR vyšších řádů. Zde je však potřeba dodat, že doba výpočtu, i když nebyla předmětem měření, byla v případě Galerkinovy (c) i kolokační (d) metody znatelně delší, než v případě Runge-Kutta metod (a) (b). Doba výpočtu v případě Galerkinovy metody byla ještě asi $5\times$ delší, než v případě kolokační metody. U obou metod za to může řešení velkých nelineárních soustav.

8. Závěr

V této práci jsme tedy rozebrali, algoritmizovali a otestovali několik metod pro numerické řešení obyčejných diferenciálních rovnic. Runge-Kutta metody se na první pohled jeví jako nejjednodušší, avšak největší výzva v jejich případě spočívá v hledání koeficientů Butcherových tabulek. Každopádně z hlediska algoritmizace jsou explicitní Runge-Kutta metody nejjednodušší na implementaci i použití.

Dále bych tedy rozebral strategie numerického řešení ODR pomocí metod uvedených v této práci. Pokud tedy máme ODR, o které předem nic nevíme, dobrou strategií bude začít ji řešit pomocí některé explicitní Runge-Kutty s malým krokem a pomocí nějakého adaptivního řízení tento krok upravovat.

Pokud víme, že budeme řešit extrémně tuhé ODR, pak se vyplatí použít implicitní Runge-Kutta metodu, kde vyřešení oněch nelineárních soustav je rychlejší než provádět řádově více kroku pomocí explicitní Runge-Kutty pro dosažení stejné přesnosti.

Pokud řešíme lineární ODR a víme, že řešení je dostatečně hladké, pak můžeme přistoupit k řešení pomocí Galerkinovy, nebo kolokační metody. V případě Galerkinovy metody je potřeba dodat, že všechny integrály musejí být vypočteny s dostatečnou přesností, což v případě výpočtů s vysokou přesností přidává další výpočetní zátěž. Na základě toho se jeví kolokace jako jasná volba, avšak je potřeba mít na paměti, že se o této metodě teoreticky moc neví a konvergence vůbec nemusí být zajištěna. Kolokační metodu bychom tedy měli brát spíše jako heuristiku.

Pokud řešíme nelineární ODR, použití Galerkinovy i kolokační metody nemusí být vhodné, i když je řešení dostatečně hladké. Měření rychlosti sice nebylo předmětem žádného numerického experimentu, i tak je potřeba zmínit, že doba řešení nelineárního problému pomocí Galerkina i kolokace je značná. To znamená, že použití Galerkina, nebo kolokace na nelineární problémy bude vhodné jen tehdy, je-li řešení dostatečně hladké a potřebné přesnosti nedosáhneme pomocí Runge-Kutta vyššího řádu ani v řádově desítkách tisíc kroků na relativně malém intervalu řešení.

Nakonec bych rád zmínil, že z výsledků testování jak v této práci i mimo, by šla zformulovat hypotéza, které by říkala pokud je řešení dostatečně hladké a Galerkinova metoda konverguje s řádem n , nebo exponenciálně, pak bude kolokace taktéž konvergovat s řádem n , nebo exponenciálně. K přesnější formulaci a případnému důkazu je potřeba další studování obou metod. Další výzkum by vyžadovaly i praktické testy metod včetně provedení optimalizované implementace a srovnání poměrů rychlosti a přesnosti pro jednotlivé metody.

Literatura

1. CODDINGTON, Earl A.; LEVINSON, Norman. *Theory of ordinary differential equations*. TMH-ed., 9. repr. New Delhi: TATA McGraw-Hill, 1987. ISBN 9780070992566. original-date: 1955.
2. BUTCHER, J.C. A history of Runge-Kutta methods. *Applied Numerical Mathematics*. 1996-03, roč. 20, č. 3, s. 247–260. ISSN 01689274. Dostupné z DOI: 10.1016/0168-9274(95)00108-5.
3. HAMMER, Preston C.; HOLLINGSWORTH, Jack W. Trapezoidal Methods of Approximating Solutions of Differential Equations. *Mathematical Tables and Other Aids to Computation*. 1955-07, roč. 9, č. 51, s. 92. ISSN 08916837. Dostupné z DOI: 10.2307/2002064.
4. ALEXANDER, Roger. Diagonally Implicit Runge–Kutta Methods for Stiff O.D.E.'s. *SIAM Journal on Numerical Analysis*. 1977-12, roč. 14, č. 6, s. 1006–1021. ISSN 0036-1429, ISSN 1095-7170. Dostupné z DOI: 10.1137/0714068.
5. *Solving Ordinary Differential Equations I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. Springer Series in Computational Mathematics. ISBN 9783540566700 9783540788621. Dostupné z DOI: 10.1007/978-3-540-78862-1.
6. STEINBACH, Olaf. *Numerical approximation methods for elliptic boundary value problems: finite and boundary elements*. New York: Springer, 2008. ISBN 9780387313122 9780387688053.
7. XIU, Dongbin. *Numerical methods for stochastic computations: a spectral method approach*. Princeton, N.J: Princeton University Press, 2010. ISBN 9780691142128. OCLC: ocn466341417.
8. PRESS, William H. (ed.). *Numerical recipes: the art of scientific computing*. 3rd ed. Cambridge, UK ; New York: Cambridge University Press, 2007. ISBN 9780521880688 9780521884075. OCLC: ocn123285342.